## APPENDIX 1

**INVEVNTORY MANAGEMENT SYSTEM**

**END TERM REPORT**

**By:**

**Dikshant Sharma(11918272)**

**Varkala Balanath(11909771*)***

**Section: K19PV**

**Roll Numbers: 59 *and* 49** *respectively*



**Department of Intelligent Systems,**

**School of Computer Science Engineering,**

**Lovely Professional University, Jalandhar**

November, 2020

## APPENDIX 2

**Student Declaration**

This is to declare that this report has been written by us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.

Name: Dikshant sharma(59),Varkala Balanath(49)

Date: 30/10/2020

**APPENDIX 3**

##### **INTRODUCTION TO INVENTORY MANAGEMENT:**

Inventory management and supplychain management are the backbone of any business operations. With the development of technology and availability of process driven software applications, inventory management has undergone revolutionary changes.In any business or organization, all functions are interlinked and connected to each other and are often overlapping. Some key aspects like supply chain management, logistics and inventory form the backbone of the business delivery function. Therefore these functions are extremely important to marketing managers as well as finance controllers. Inventory management is a very important function that determines the health of the supply chain as well as the impacts the financial health of the balance sheet. Every organization constantly strives to maintain optimum inventory to be able to meet its requirements and avoid over or under inventory that can impact the financial figures. Inventory is always dynamic. Inventory management requires constant and careful evaluation of external and internal factors and control through planning and review. Most of the organizations have a separate department or job function called inventory planners who continuously monitor, control and review inventory and interface with production, procurement and finance departments.

##### **OBJECTIVE:**

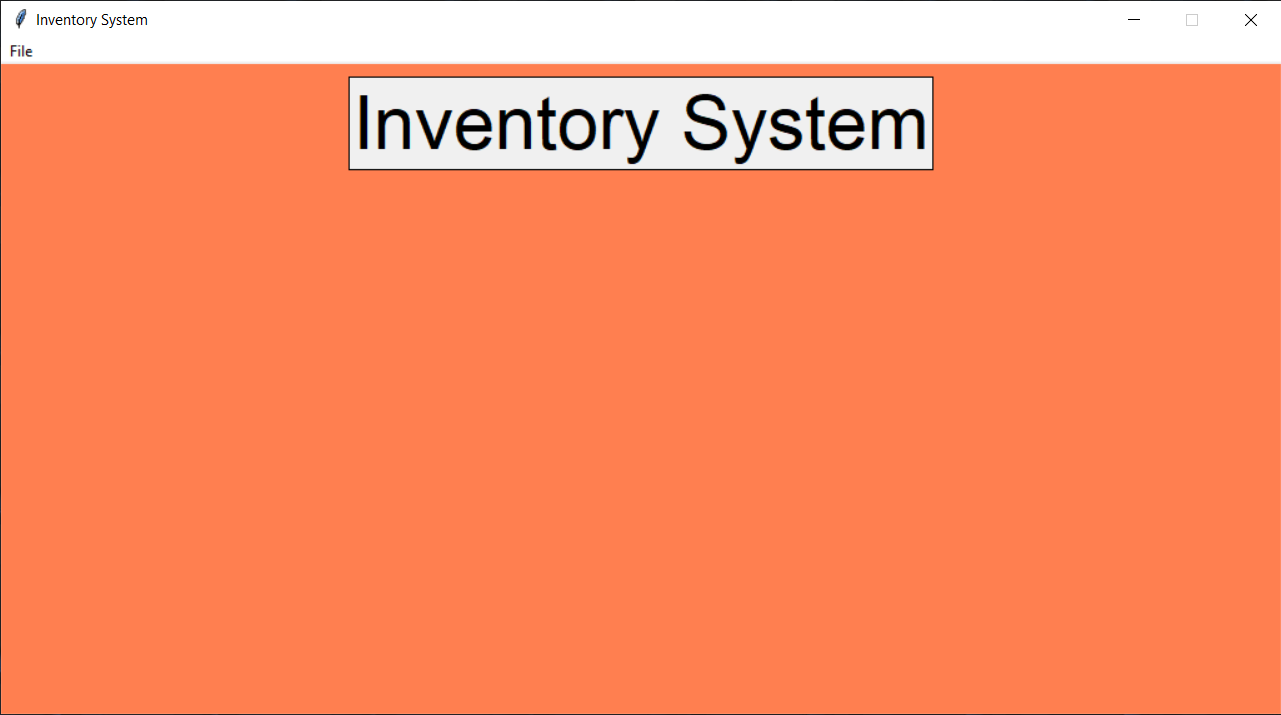
The objective of this project is to program a inventory management system for a shop using Python and Python libraries which can perform all the basic task in a inventory system and user can easily use it and provide a user friendly experience.

##### **SOFTWARE REQUIREMENTS:**

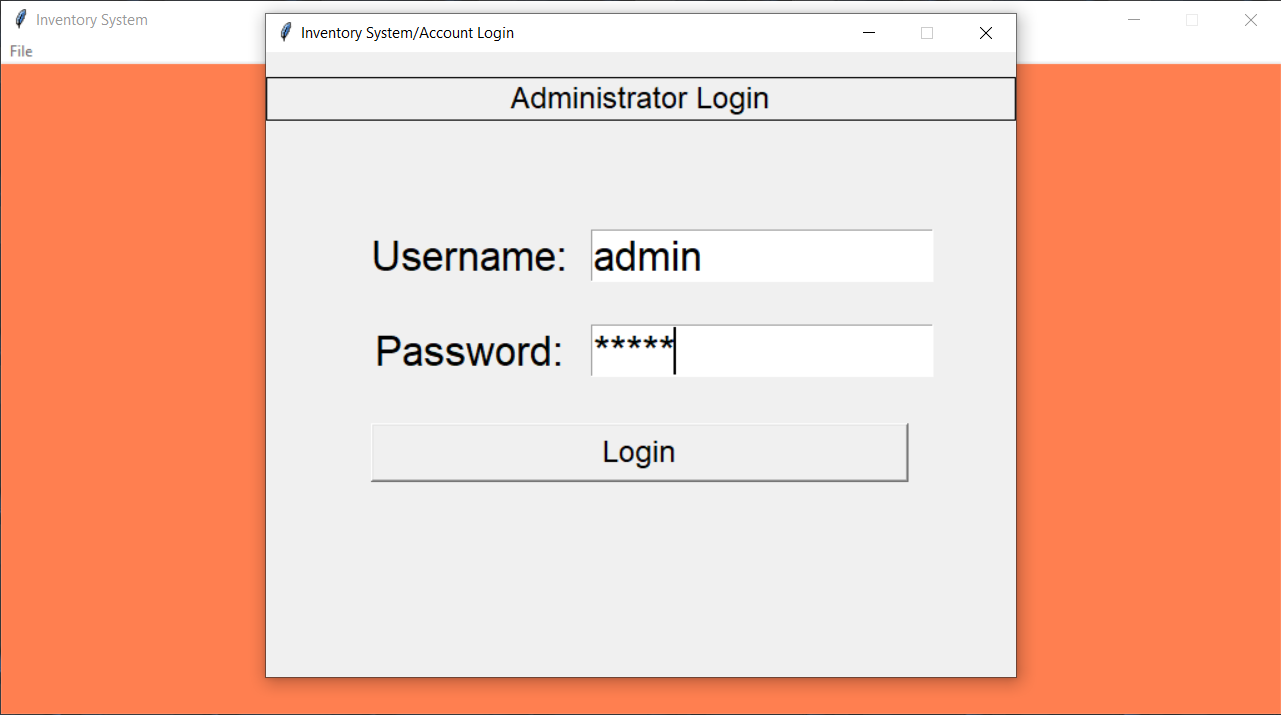
* Visual Studio code
* Jupiter Notebook
* SQL Server
* Windows, Linux, mac

##### **PROJECT SCREENSHOTS:**

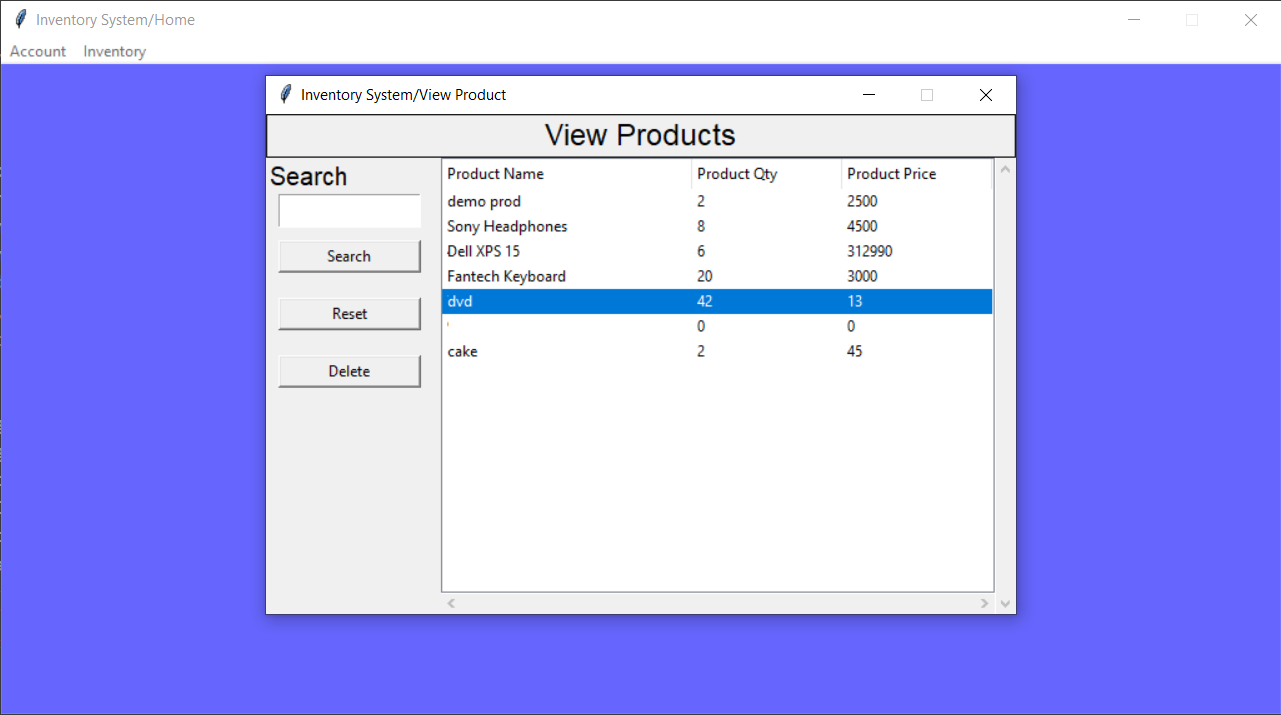
This is the home page of the inventory system



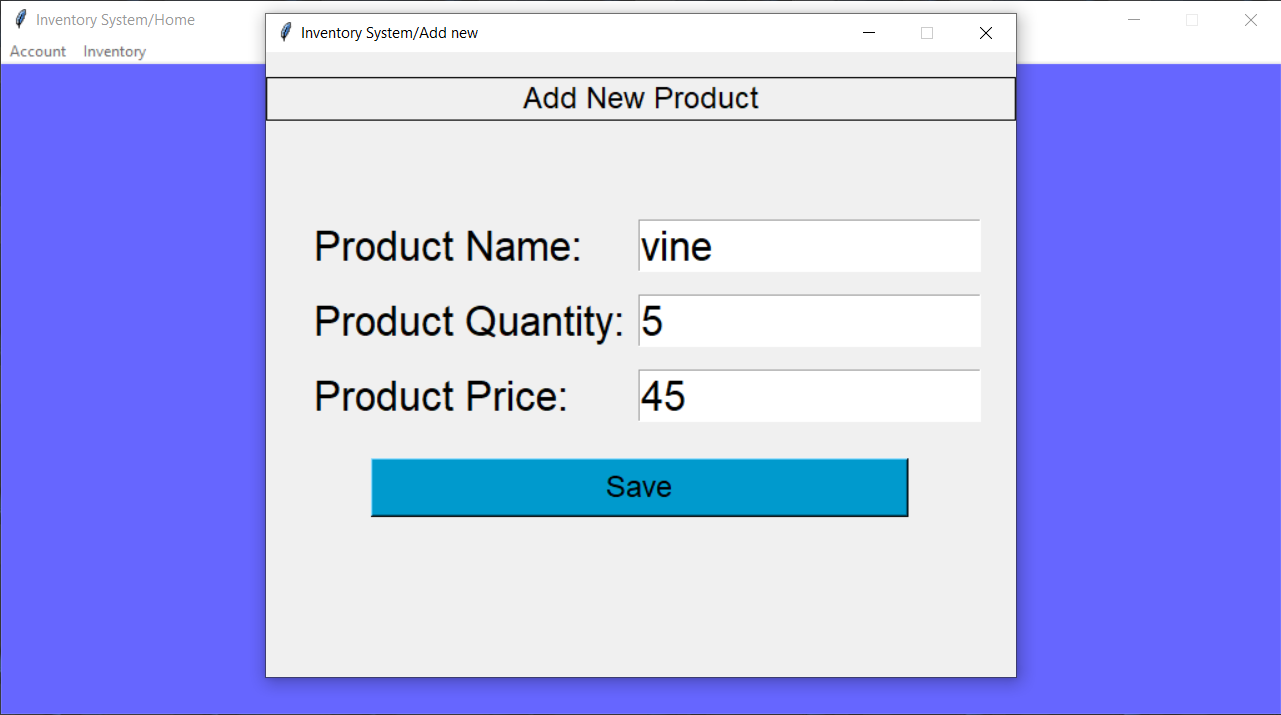
* This is the login page of the inventory system and a user can login using the user id as ”***admin***” and password as ”***admin***” and without this user can’t access into the system.



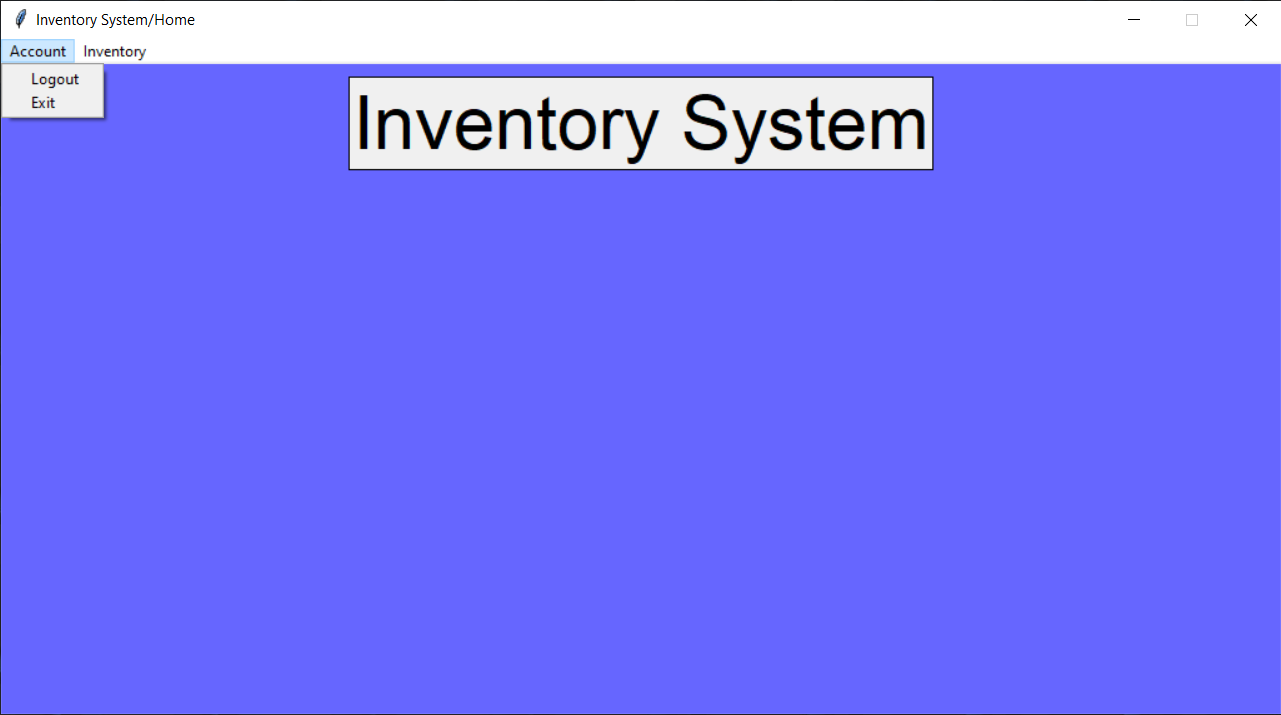
* This is the view section in which a user can view the list of items and can easily search or traverse through the list of items and can also delete the item if the item is not required.



This is the Add new product Section in which a user can Add a new product and add the quantity and the price of the product and this data will get store in the database through the Sql server.



Finally the user can exit the software using exit button and a message will be displays at the time of logging out form the system conferming the user wether he want to exit the system.



##### **PROJECT SOURCE CODE:**

from tkinter import \*

import tkinter.messagebox as tkMessageBox

import sqlite3

import tkinter.ttk as ttk

root = Tk()

root.title("Inventory System")

width = 1024

height = 520

screen\_width = root.winfo\_screenwidth()

screen\_height = root.winfo\_screenheight()

x = (screen\_width/2) - (width/2)

y = (screen\_height/2) - (height/2)

root.geometry("%dx%d+%d+%d" % (width, height, x, y))

root.resizable(0, 0)

root.config(bg="coral")

USERNAME = StringVar()

PASSWORD = StringVar()

PRODUCT\_NAME = StringVar()

PRODUCT\_PRICE = IntVar()

PRODUCT\_QTY = IntVar()

SEARCH = StringVar()

def Database():

    global conn, cursor

    conn = sqlite3.connect("pythontut.db")

    cursor = conn.cursor()

    cursor.execute("CREATE TABLE IF NOT EXISTS `admin` (admin\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, username TEXT, password TEXT)")

    cursor.execute("CREATE TABLE IF NOT EXISTS `product` (product\_id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, product\_name TEXT, product\_qty TEXT, product\_price TEXT)")

    cursor.execute("SELECT \* FROM `admin` WHERE `username` = 'admin' AND `password` = 'admin'")

    if cursor.fetchone() is None:

        cursor.execute("INSERT INTO `admin` (username, password) VALUES('admin', 'admin')")

        conn.commit()

def Exit():

    result = tkMessageBox.askquestion('Inventory System', 'Are you sure you want to exit?', icon="warning")

    if result == 'yes':

        root.destroy()

        exit()

def Exit2():

    result = tkMessageBox.askquestion('Inventory System', 'Are you sure you want to exit?', icon="warning")

    if result == 'yes':

        Home.destroy()

        exit()

def ShowLoginForm():

    global loginform

    loginform = Toplevel()

    loginform.title("Inventory System/Account Login")

    width = 600

    height = 500

    screen\_width = root.winfo\_screenwidth()

    screen\_height = root.winfo\_screenheight()

    x = (screen\_width/2) - (width/2)

    y = (screen\_height/2) - (height/2)

    loginform.resizable(0, 0)

    loginform.geometry("%dx%d+%d+%d" % (width, height, x, y))

    LoginForm()

def LoginForm():

    global lbl\_result

    TopLoginForm = Frame(loginform, width=600, height=100, bd=1, relief=SOLID)

    TopLoginForm.pack(side=TOP, pady=20)

    lbl\_text = Label(TopLoginForm, text="Administrator Login", font=('arial', 18), width=600)

    lbl\_text.pack(fill=X)

    MidLoginForm = Frame(loginform, width=600)

    MidLoginForm.pack(side=TOP, pady=50)

    lbl\_username = Label(MidLoginForm, text="Username:", font=('arial', 25), bd=18)

    lbl\_username.grid(row=0)

    lbl\_password = Label(MidLoginForm, text="Password:", font=('arial', 25), bd=18)

    lbl\_password.grid(row=1)

    lbl\_result = Label(MidLoginForm, text="", font=('arial', 18))

    lbl\_result.grid(row=3, columnspan=2)

    username = Entry(MidLoginForm, textvariable=USERNAME, font=('arial', 25), width=15)

    username.grid(row=0, column=1)

    password = Entry(MidLoginForm, textvariable=PASSWORD, font=('arial', 25), width=15, show="\*")

    password.grid(row=1, column=1)

    btn\_login = Button(MidLoginForm, text="Login", font=('arial', 18), width=30, command=Login)

    btn\_login.grid(row=2, columnspan=2, pady=20)

    btn\_login.bind('<Return>', Login)

def Home():

    global Home

    Home = Tk()

    Home.title("Inventory System/Home")

    width = 1024

    height = 520

    screen\_width = Home.winfo\_screenwidth()

    screen\_height = Home.winfo\_screenheight()

    x = (screen\_width/2) - (width/2)

    y = (screen\_height/2) - (height/2)

    Home.geometry("%dx%d+%d+%d" % (width, height, x, y))

    Home.resizable(0, 0)

    Title = Frame(Home, bd=1, relief=SOLID)

    Title.pack(pady=10)

    lbl\_display = Label(Title, text="Inventory System", font=('arial', 45))

    lbl\_display.pack()

    menubar = Menu(Home)

    filemenu = Menu(menubar, tearoff=0)

    filemenu2 = Menu(menubar, tearoff=0)

    filemenu.add\_command(label="Logout", command=Logout)

    filemenu.add\_command(label="Exit", command=Exit2)

    filemenu2.add\_command(label="Add new", command=ShowAddNew)

    filemenu2.add\_command(label="View", command=ShowView)

    menubar.add\_cascade(label="Account", menu=filemenu)

    menubar.add\_cascade(label="Inventory", menu=filemenu2)

    Home.config(menu=menubar)

    Home.config(bg="#6666ff")

def ShowAddNew():

    global addnewform

    addnewform = Toplevel()

    addnewform.title("Inventory System/Add new")

    width = 600

    height = 500

    screen\_width = Home.winfo\_screenwidth()

    screen\_height = Home.winfo\_screenheight()

    x = (screen\_width/2) - (width/2)

    y = (screen\_height/2) - (height/2)

    addnewform.geometry("%dx%d+%d+%d" % (width, height, x, y))

    addnewform.resizable(0, 0)

    AddNewForm()

def AddNewForm():

    TopAddNew = Frame(addnewform, width=600, height=100, bd=1, relief=SOLID)

    TopAddNew.pack(side=TOP, pady=20)

    lbl\_text = Label(TopAddNew, text="Add New Product", font=('arial', 18), width=600)

    lbl\_text.pack(fill=X)

    MidAddNew = Frame(addnewform, width=600)

    MidAddNew.pack(side=TOP, pady=50)

    lbl\_productname = Label(MidAddNew, text="Product Name:", font=('arial', 25), bd=10)

    lbl\_productname.grid(row=0, sticky=W)

    lbl\_qty = Label(MidAddNew, text="Product Quantity:", font=('arial', 25), bd=10)

    lbl\_qty.grid(row=1, sticky=W)

    lbl\_price = Label(MidAddNew, text="Product Price:", font=('arial', 25), bd=10)

    lbl\_price.grid(row=2, sticky=W)

    productname = Entry(MidAddNew, textvariable=PRODUCT\_NAME, font=('arial', 25), width=15)

    productname.grid(row=0, column=1)

    productqty = Entry(MidAddNew, textvariable=PRODUCT\_QTY, font=('arial', 25), width=15)

    productqty.grid(row=1, column=1)

    productprice = Entry(MidAddNew, textvariable=PRODUCT\_PRICE, font=('arial', 25), width=15)

    productprice.grid(row=2, column=1)

    btn\_add = Button(MidAddNew, text="Save", font=('arial', 18), width=30, bg="#009ACD", command=AddNew)

    btn\_add.grid(row=3, columnspan=2, pady=20)

def AddNew():

    Database()

    cursor.execute("INSERT INTO `product` (product\_name, product\_qty, product\_price) VALUES(?, ?, ?)", (str(PRODUCT\_NAME.get()), int(PRODUCT\_QTY.get()), int(PRODUCT\_PRICE.get())))

    conn.commit()

    PRODUCT\_NAME.set("")

    PRODUCT\_PRICE.set("")

    PRODUCT\_QTY.set("")

    cursor.close()

    conn.close()

def ViewForm():

    global tree

    TopViewForm = Frame(viewform, width=600, bd=1, relief=SOLID)

    TopViewForm.pack(side=TOP, fill=X)

    LeftViewForm = Frame(viewform, width=600)

    LeftViewForm.pack(side=LEFT, fill=Y)

    MidViewForm = Frame(viewform, width=600)

    MidViewForm.pack(side=RIGHT)

    lbl\_text = Label(TopViewForm, text="View Products", font=('arial', 18), width=600)

    lbl\_text.pack(fill=X)

    lbl\_txtsearch = Label(LeftViewForm, text="Search", font=('arial', 15))

    lbl\_txtsearch.pack(side=TOP, anchor=W)

    search = Entry(LeftViewForm, textvariable=SEARCH, font=('arial', 15), width=10)

    search.pack(side=TOP,  padx=10, fill=X)

    btn\_search = Button(LeftViewForm, text="Search", command=Search)

    btn\_search.pack(side=TOP, padx=10, pady=10, fill=X)

    btn\_reset = Button(LeftViewForm, text="Reset", command=Reset)

    btn\_reset.pack(side=TOP, padx=10, pady=10, fill=X)

    btn\_delete = Button(LeftViewForm, text="Delete", command=Delete)

    btn\_delete.pack(side=TOP, padx=10, pady=10, fill=X)

    scrollbarx = Scrollbar(MidViewForm, orient=HORIZONTAL)

    scrollbary = Scrollbar(MidViewForm, orient=VERTICAL)

    tree = ttk.Treeview(MidViewForm, columns=("ProductID", "Product Name", "Product Qty", "Product Price"), selectmode="extended", height=100, yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)

    scrollbary.config(command=tree.yview)

    scrollbary.pack(side=RIGHT, fill=Y)

    scrollbarx.config(command=tree.xview)

    scrollbarx.pack(side=BOTTOM, fill=X)

    tree.heading('ProductID', text="ProductID",anchor=W)

    tree.heading('Product Name', text="Product Name",anchor=W)

    tree.heading('Product Qty', text="Product Qty",anchor=W)

    tree.heading('Product Price', text="Product Price",anchor=W)

    tree.column('#0', stretch=NO, minwidth=0, width=0)

    tree.column('#1', stretch=NO, minwidth=0, width=0)

    tree.column('#2', stretch=NO, minwidth=0, width=200)

    tree.column('#3', stretch=NO, minwidth=0, width=120)

    tree.column('#4', stretch=NO, minwidth=0, width=120)

    tree.pack()

    DisplayData()

def DisplayData():

    Database()

    cursor.execute("SELECT \* FROM `product`")

    fetch = cursor.fetchall()

    for data in fetch:

        tree.insert('', 'end', values=(data))

    cursor.close()

    conn.close()

def Search():

    if SEARCH.get() != "":

        tree.delete(\*tree.get\_children())

        Database()

        cursor.execute("SELECT \* FROM `product` WHERE `product\_name` LIKE ?", ('%'+str(SEARCH.get())+'%',))

        fetch = cursor.fetchall()

        for data in fetch:

            tree.insert('', 'end', values=(data))

        cursor.close()

        conn.close()

def Reset():

    tree.delete(\*tree.get\_children())

    DisplayData()

    SEARCH.set("")

def Delete():

    if not tree.selection():

       print("ERROR")

    else:

        result = tkMessageBox.askquestion('Inventory System', 'Are you sure you want to delete this record?', icon="warning")

        if result == 'yes':

            curItem = tree.focus()

            contents =(tree.item(curItem))

            selecteditem = contents['values']

            tree.delete(curItem)

            Database()

            cursor.execute("DELETE FROM `product` WHERE `product\_id` = %d" % selecteditem[0])

            conn.commit()

            cursor.close()

            conn.close()

def ShowView():

    global viewform

    viewform = Toplevel()

    viewform.title("Inventory System/View Product")

    width = 600

    height = 400

    screen\_width = Home.winfo\_screenwidth()

    screen\_height = Home.winfo\_screenheight()

    x = (screen\_width/2) - (width/2)

    y = (screen\_height/2) - (height/2)

    viewform.geometry("%dx%d+%d+%d" % (width, height, x, y))

    viewform.resizable(0, 0)

    ViewForm()

def Logout():

    result = tkMessageBox.askquestion('Inventory System', 'Are you sure you want to logout?', icon="warning")

    if result == 'yes':

        admin\_id = ""

        root.deiconify()

        Home.destroy()

def Login(event=None):

    global admin\_id

    Database()

    if USERNAME.get == "" or PASSWORD.get() == "":

        lbl\_result.config(text="Please complete the required field!", fg="red")

    else:

        cursor.execute("SELECT \* FROM `admin` WHERE `username` = ? AND `password` = ?", (USERNAME.get(), PASSWORD.get()))

        if cursor.fetchone() is not None:

            cursor.execute("SELECT \* FROM `admin` WHERE `username` = ? AND `password` = ?", (USERNAME.get(), PASSWORD.get()))

            data = cursor.fetchone()

            admin\_id = data[0]

            USERNAME.set("")

            PASSWORD.set("")

            lbl\_result.config(text="")

            ShowHome()

        else:

            lbl\_result.config(text="Invalid username or password", fg="red")

            USERNAME.set("")

            PASSWORD.set("")

    cursor.close()

    conn.close()

def ShowHome():

    root.withdraw()

    Home()

    loginform.destroy()

menubar = Menu(root)

filemenu = Menu(menubar, tearoff=0)

filemenu.add\_command(label="Account", command=ShowLoginForm)

filemenu.add\_command(label="Exit", command=Exit)

menubar.add\_cascade(label="File", menu=filemenu)

root.config(menu=menubar)

Title = Frame(root, bd=1, relief=SOLID)

Title.pack(pady=10)

lbl\_display = Label(Title, text="Inventory System", font=('arial', 45))

lbl\_display.pack()

if \_\_name\_\_ == '\_\_main\_\_':

    root.mainloop()

##### **TECHNOLOGIES & FRAMEWORKS USED.**

* **Python**
* **Tkinter for GUI**
* **SQL database**

GITHUB LINK:

[**https://github.com/dikshantsharma12/Inventory-management-system.git**](https://github.com/dikshantsharma12/Inventory-management-system.git)

##### **REFERENCES:**

* [**https://www.w3schools.com/python/default.asp**](https://www.w3schools.com/python/default.asp)
* [**https://stackoverflow.com/search?q=inventory+management+system&s=06b704eb-bb77-4533-bf09-8f89c7f9e9d5**](https://stackoverflow.com/search?q=inventory+management+system&s=06b704eb-bb77-4533-bf09-8f89c7f9e9d5)
* [**https://projectnotes.org/it-projects/inventory-management-system-in-python-with-source-code/**](https://projectnotes.org/it-projects/inventory-management-system-in-python-with-source-code/)
* [**https://github.com/JaishreeJaishankar/Stock-Management-System**](https://github.com/JaishreeJaishankar/Stock-Management-System)
* [**https://youtube.com/**](https://youtube.com/)

##### **MODULES:**

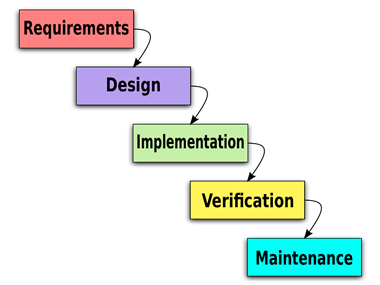
The system comprises of 1 major modules with their sub-modules as follows:

**User:**

* + **Login:** User can login his account using id and password.
  + **Add Product:** User can add product.
  + **View list:** User can also view the list of items.
  + **Inventory:** User can add Inventory.
  + **Update stock:** User can add, delete the stock.
  + **Search product:** User can search the Available product.

###### **DESCRIPTION**

The waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach that was used for software development.



**ADMIN**

**APPENDIX 4**

BONAFIDE CERTIFICATE

Certified that this project report “Inventory Management” is the Bonafide work of “Dikshant sharma and Varkala Balanath” who carried out the project work under my supervision.

### <<Signature of the Supervisor>>(Due to Covid 19, signature is exempted )

<<Name of supervisor>>

<<Academic Designation>>

<<ID of Supervisor>>

<<Department of Supervisor>>